选择器实验报告

PB16030899 朱河勤

**一、实验目的**

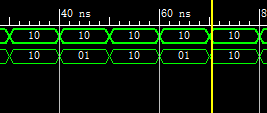
掌握选择器的编码原理以及设计、调试方法，学会用选择器实现组合电路。以及verilog的实例化。

**二、实验内容**

* 参照MUX2，画出1bit MUX4电路结构图
* 将电路结构图转换成Verilog代码实现
* 用2个1bit MUX4采用模块例化方式，实现2bit位宽MUX4
* 利用设计好的2bit位宽MUX4实现异或、同或逻辑
  + o[1] = sel[1] *xor* sel[0] o[0] = sel[1] *xnor* sel[0]
* 对用MUX4实现的同或、异或门进行仿真

用开关做为输入、LED做为输出

**三、仿真结果**

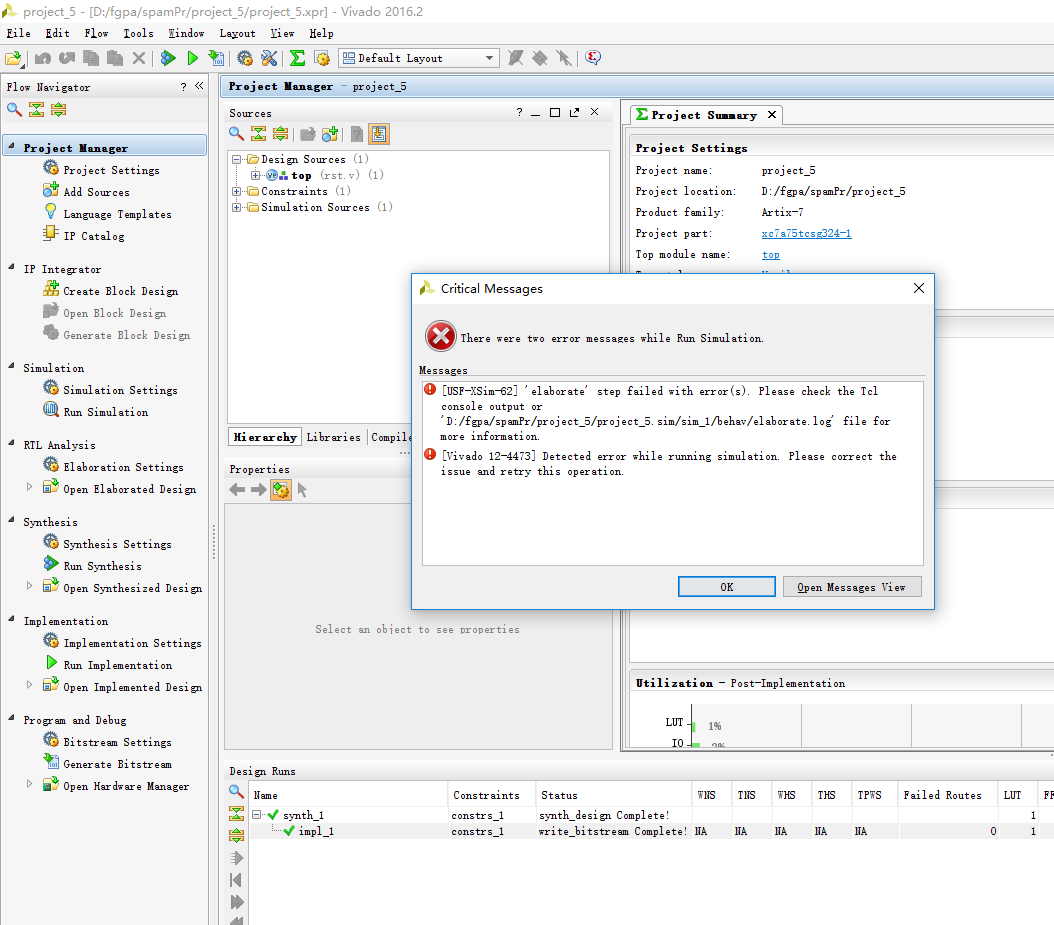


输入： 00 01 10 11 //两位分别是选择输入sel0，sel1

输出： 01 10 10 01 //两位分别是xor，xnor

**四．实验分析**

1. 首先要熟悉verilog 语法，弄清楚仿真文件的写法以及实例化的方法。  
 我写代码写出来比较快，可是仿真的时候出了问题。



2. 如上图所示，在网上也查不到解决方法。问了助教，助教也不知道怎么回事。问了老师后，重新复制项目，然后可以仿真。但是再次仿真还是不行。目前的解决方法是每仿真一次就复制项目一次。

3. 写bit文件时还算成功，只要分清楚io，选择好管脚，就行了。

**五．意见建议**老师，和助教指导得很好。自己也应该多学多练，只是这方面的资料太少，而且刚开始入门，不熟悉verilog 以及vivado，才会踩很多坑。而且网上也难以找到一些问题的解决方法。只能自己摸索。

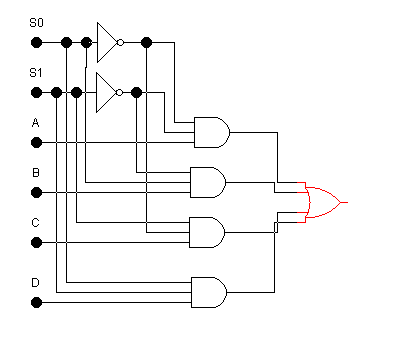
**六、实验总结**

通过本实验，了解了选择器逻辑功能和接线方法；初步学习了Verilog HDL硬件描述语言的基本语法。

为保证电路正确，连线后可以在主要的逻辑步骤，加入一些额外输出，以检查电路各部分逻辑的正确性。一定得熟悉verilog语法，以避免浪费太多时间

**七．附录**

源代图及源代码如下：

****

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2017/10/26 20:13:25

// Design Name:

// Module Name: top

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module mux4(

input [3:0] D, [1:0] S,

output reg Y

);

always @

begin

case(S)

2'd0:Y = D[0];

2'd1:Y = D[1];

2'd2:Y = D[2];

2'd3:Y = D[3];

endcase

end

endmodule

module mux4\_2(

input [7:0] i,

input [1:0] s,

output [1:0] o);

mux4 xor\_z(

.D(i[3:0]),

.S(s[1:0]),

.Y(out[0]));

mux4 xnor\_z(

.D(i[7:4]),

.S(s[1:0]),

.Y(o[1]));

endmodule

module top(

inuput [1:0] sel,

output [1:0] out

);

mux4\_2 rst(

.i(2'b0110\_1001),

.s(sel),

.o(out)

);

endmodule